# PROPX Fair and Efficient Allocation of Indivisible Chores


H. Aziz, B. Li, H. Moulin, X. Wu, and X. Zhu

Computer Science and Engineering School, UNSW Sydney, Australia


## Motivations and Questions

| | Vacuum | Laundry | Wash Dishes | Trash to Curb |
|---|---|---|---|---|
| Alice | −0.15 | −0.45 | −0.27 | −0.13 |
| Bob | −0.50 | −0.13 | −0.35 | −0.02 |
| Celine | −0.68 | −0.07 | −0.05 | −0.20 |

- Four indivisible chores to be fully allocated to three people.
- Each person has their own valuation on the chores.

Question: How do we allocate these indivisible chores fairly and efficiently?

**[Fairness Concept]** Proportional up to any item (PROPX): Every agent $i$'s utility is at least some portion of $u_i(O)$ after removing one chore.

**[Efficiency Concept]** Pareto Optimal (PO): No other allocations can strictly increase some clients' utility without decreasing any other's utility.

## Pareto Improvements that Preserve PROPX (resolving trading cycles)

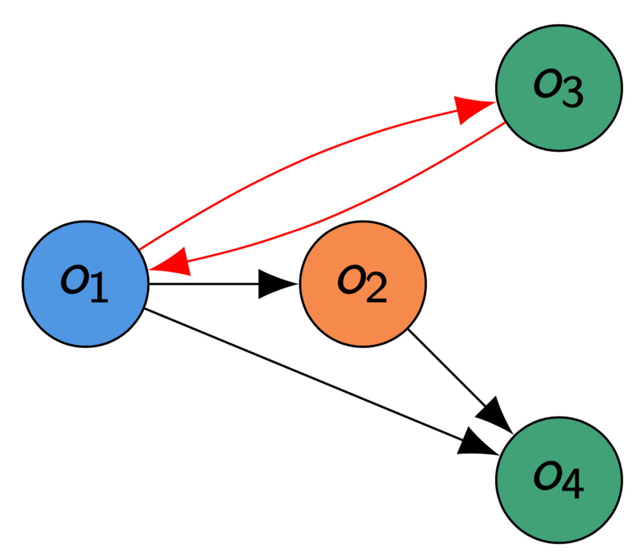
**Trading graph**: $G(X)$ of an allocation $X$.
**Vertices**: Each chore in $O$ is a vertex.
**Edges**: For any two vertices $o$ and $o'$, there is a directed edge from $o$ to $o'$ if $u_i(o') \geq u_i(o)$ where $o \in X_i$ and $o' \notin X_i$. The edge is strict if $u_i(o') > u_i(o)$.
**Trading cycle**: a cycle $C$ in $G(X)$ containing at least one strict edge.
**Resolving a trading cycle**: Allocation $Y$ is a result of resolving trading cycle $C$ if for each edge $(o, o') \in C$, it holds that

$$Y_i = (X_i \setminus \{o\}) \cup \{o'\}.$$

(Each agent involved in a trading cycle gives away a chore they hate more and receives a chore they hate less.)

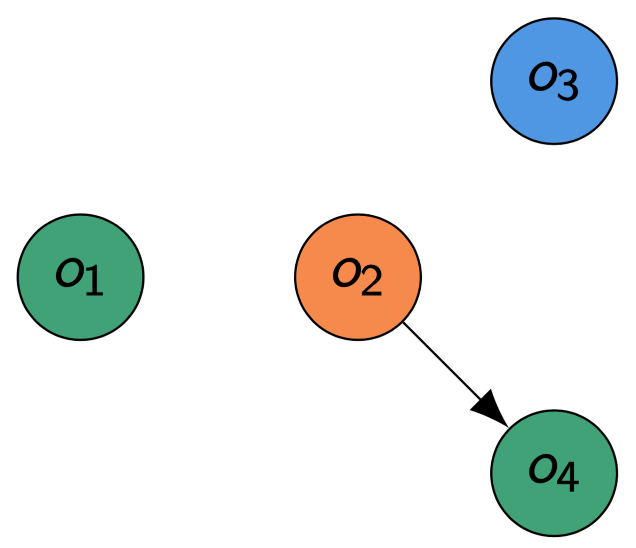
| | $o_1$ | $o_2$ | $o_3$ | $o_4$ |
|---|---|---|---|---|
| 1 | −0.15 | −0.45 | −0.27 | −0.13 |
| 2 | −0.50 | −0.13 | −0.35 | −0.02 |
| 3 | −0.68 | −0.07 | −0.05 | −0.20 |

Table: allocation $X$




| | $o_1$ | $o_2$ | $o_3$ | $o_4$ |
|---|---|---|---|---|
| 1 | −0.15 | −0.45 | −0.27 | −0.13 |
| 2 | −0.50 | −0.13 | −0.35 | −0.02 |
| 3 | −0.68 | −0.07 | −0.05 | −0.20 |

Table: allocation $Y$




## Our Model

- $m$ indivisible chores in set $O$.
- $n$ asymmetric agents in set $N$, each $i \in N$ has a weight $b_i > 0$ and $\sum_{i \in N} b_i = 1$.
- Each agent $i \in N$ has an additive utility function $u_i: 2^O \to \mathbb{R}^- \cup \{0\}$.
- An allocation $X = (X_1, \ldots, X_n)$ where $X_i$ is the allocated set of chores to agent $i$.

## Our Approach

- Compute a PROPX allocation in polynomial time.
- Given a PROPX allocation, perform a series of Pareto improvement (resolving trading cycles) over it that preserve PROPX until it is PO.

## Our Results

- The process of resolving trading cycles will terminate in polynomial time.
- [Our approach may not work] Starting with a PROPX allocation, the process of resolving trading cycles may terminate before reaching a PO allocation.
- [PROPX and PO with more restrictions to make our approach work] To have restricted utility functions:

  An allocation $X$ is not PO with respect to

  1. lexicographic preferences

  2. bivalued preferences

  if and only if there exists a trading cycle in $G(X)$.
- For lexicographic and bivalued utilities, there exists a polynomial-time algorithm that computes an allocation that is both PROPX and PO even for asymmetric weights.